

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-198362  
(43)Date of publication of application : 11.07.2003

(51)Int.Cl. H03K 19/177  
G06F 9/40  
G06F 9/54

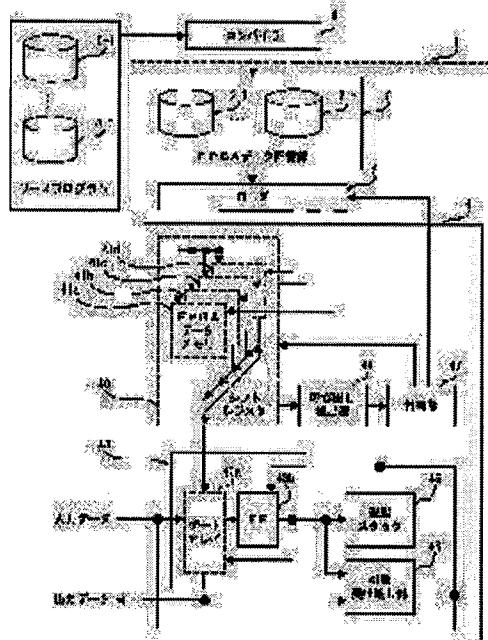
(21)Application number : 2001-401462 (71)Applicant : TOKYO ELECTRON DEVICE LTD  
NISHIHARA AKINORI  
(22)Date of filing : 28.12.2001 (72)Inventor : MITA TAKASHI  
NISHIHARA AKINORI

## (54) OPERATING SYSTEM

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To realize efficient execution of a large scale program comprising a plurality of program modules directly by means of hardware without using a general purpose CPU.

**SOLUTION:** A gate array 43 performs operation by means of hardware according to an FPGA data module stored in an FPGA data memory specified in a shift register 40. When a call detecting section 44 detects that a module stored in the FPGA data memory calls other module, data of intermediate operational results stored in a flip-flop 43b is retreated into a retreat stack 45 and an argument being delivered to a called module is stored temporarily in an argument delivery section 46. When an FPGA data memory storing the called module is specified, subsequently, to the shift register 40 and reset to a calling module, data retreated into the retreat stack 45 is rewritten to the flip-flop 43b.



## LEGAL STATUS

[Date of request for examination] 28.01.2002

[Date of sending the examiner's  
decision of rejection]

[Kind of final disposal of application  
other than the examiner's decision  
of rejection or application  
converted registration]

[Date of final disposal for  
application]

[Patent number] 3540796

[Date of registration] 02.04.2004

[Number of appeal against  
examiner's decision of rejection]

[Date of requesting appeal against  
examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-198362

(P2003-198362A)

(43)公開日 平成15年7月11日 (2003.7.11)

(51)Int.Cl.<sup>7</sup>  
H 03 K 19/177  
G 06 F 9/40  
9/54

識別記号  
3 1 0

F I  
H 03 K 19/177  
G 06 F 9/40  
9/06

デマコート<sup>\*</sup>(参考)  
5 B 0 3 3  
3 1 0 B 5 B 0 7 6  
6 4 0 B 5 J 0 4 2

審査請求 有 請求項の数 9 O L (全 13 頁)

(21)出願番号 特願2001-401462(P2001-401462)

(22)出願日 平成13年12月28日 (2001.12.28)

(71)出願人 500323188  
東京エレクトロンデバイス株式会社  
神奈川県横浜市都筑区東方町1番地  
(71)出願人 501186977  
西原 明法  
神奈川県川崎市中原区今井仲町374-4-505  
(72)発明者 三田 高司  
神奈川県横浜市都筑区東方町1番地 東京  
エレクトロンデバイス株式会社内  
(74)代理人 100095407  
弁理士 木村 满

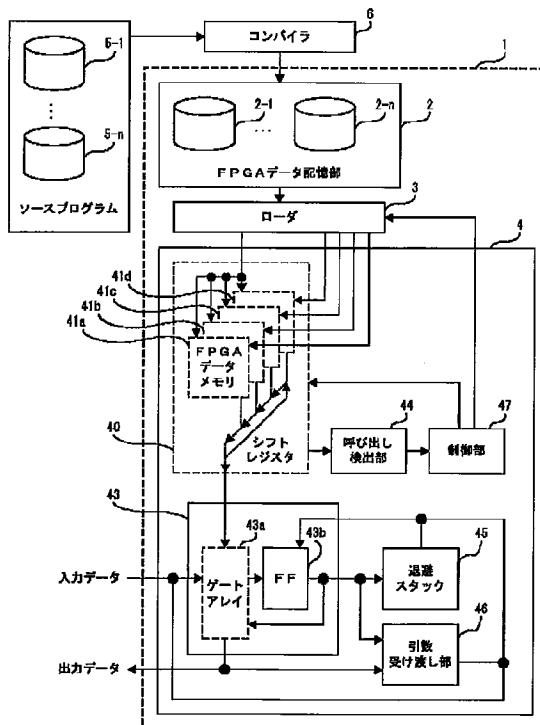
最終頁に続く

(54)【発明の名称】 演算システム

(57)【要約】

【課題】 汎用のC P Uを用いることなく、複数のプログラムモジュールからなる大規模プログラムの効率的な実行をハードウェアで直接的に実現する。

【解決手段】 ゲートアレイ43は、シフトレジスタ40に指定されたF P G Aデータメモリに記憶されたF P G Aデータモジュールに従ってハードウェア的に演算を行う。このF P G Aデータメモリに記憶されたモジュールが他のモジュールを呼び出すものであることが呼び出し検出部44によって検出されると、フリップフロップ43bに保持された演算の途中結果のデータが退避スタック45に退避され、呼び出し先のモジュールに渡す引数が引数受け渡し部46に一時保存される。その後、呼び出し先のモジュールを記憶したF P G Aデータメモリをシフトレジスタ40に指定させ、呼び出し元のモジュールに復帰する際には、退避スタック45に退避されたデータがフリップフロップ43bに書き戻される。



**【特許請求の範囲】**

【請求項1】ロックアップテーブルを含んだ複数のプログラムモジュールを各々記憶する複数のプログラム記憶手段と、各前記プログラム記憶手段のいずれかを選択する選択手段と、複数の論理回路を含み、前記選択手段に選択されたプログラム記憶手段に記憶されているプログラムモジュール中の命令に従った信号を前記複数の論理回路の1以上に入力することで、該プログラムモジュールに応じた演算を実行する論理演算手段と、を備えることを特徴とする演算システム。

【請求項2】前記選択手段は、シフトレジスタより構成されている、

ことを特徴とする請求項1に記載の演算システム。

【請求項3】各前記プログラム記憶手段は、プログラムモジュールを各々書き換え可能に記憶するものであって、

前記選択手段は、所定のプログラム記憶手段が記憶するプログラムモジュールを選択するものであって、各前記プログラムモジュールを、各前記プログラム記憶手段相互間でサイクリックに転送させる、

ことを特徴とする請求項1又は2に記載の演算システム。

【請求項4】各前記プログラム記憶手段は、プログラムモジュールを各々書き換え可能に記憶するものであって、

前記プログラム記憶手段にプログラムモジュールをロードするロード手段を更に備える、

ことを特徴とする請求項1、2又は3に記載の演算システム。

【請求項5】前記論理演算手段の内部状態を退避する退避手段と、

所定の条件が成立した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、他のプログラムモジュールを前記ロード手段にロードさせ、該他のプログラムモジュールを記憶したプログラム記憶手段を前記選択手段に選択させ、該他のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、元のプログラムモジュールに応じた演算の実行に復帰させる制御手段と、

を更に備えることを特徴とする請求項4に記載の演算システム。

【請求項6】前記複数のプログラムモジュールのうちの少なくとも一部のプログラムモジュールは、他のプログラムモジュールを呼び出す機能を含み、

前記論理演算手段で演算を実行しているプログラムモジュール中の命令における他のプログラムモジュールの呼び出しを検出する呼び出し検出手段をさらに備え、

前記制御手段は、前記呼び出し検出手段が他のプログラムモジュールの呼び出しを検出した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、呼び出し先のプログラムモジュールが前記プログラム記憶手段に記憶されていないときは該呼び出し先のプログラムモジュールを前記ロード手段にロードさせ、該呼び出し先のプログラムモジュールを記憶しているプログラム記憶手段を前記選択手段に選択させ、該呼び出し先のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、呼び出し元のプログラムモジュールに応じた演算の実行に復帰させることを特徴とする請求項5に記載の演算システム。

【請求項7】前記制御手段によって実行が切り替えられるプログラムモジュール間において引数を受け渡すための引数受け渡し手段をさらに備えることを特徴とする請求項5又は6に記載の演算システム。

【請求項8】前記退避手段は、先入れ後出し方式のスタックによって構成されることを特徴とする請求項5、6又は7に記載の演算システム。

【請求項9】前記プログラム記憶手段に記憶された各プログラムモジュールは、前記論理演算手段を構成する論理回路に入力する信号に応じたコードによって構成されていることを特徴とする請求項1乃至8のいずれか1項に記載の演算システム。

**【発明の詳細な説明】**

**【0001】**

**【発明の属する技術分野】**本発明は、プログラムの実行をハードウェアで直接的に実現できる演算システムに関し、特に大規模プログラムの実行に適した演算システムに関する。

**【0002】**

**【従来の技術】**現在の汎用コンピュータは、C P U (Central Processing Unit) がメモリに記憶されたプログラム中の命令を順次解釈しながら、演算を進めていく。C P Uは、プログラムで実行すべき演算をソフトウェアで実現するものであり、必ずしもその演算に対して最適なハードウェア構成となっていないため、最終的な演算結果を得るまでに多くのオーバーヘッドが存在する。

【0003】これに対して、プログラムの実行をハードウェアで直接的に実現するための技術として、例えば、特表平8-504285号公報(国際公開WO 94/10627号公報)や特表2000-516418号公報(国際公開WO 98/08306号公報)に示されているような、フィールドプログラマブルゲートアレイ(FPGA)を利用した演算システムが知られている。

【0004】F P G Aは、プログラムとして論理データを与えることで論理回路間の結線論理を変更し、これによつてハードウェア的に演算結果を得ることをできるようにしたものである。F P G Aを利用して演算を行うこ

とによって、特定の演算専用に構成されたハードウェア回路ほどは高速ではないが、従来の汎用コンピュータのようなC P Uによる演算に比べると、非常に高速に演算結果を得ることができる。

#### 【0005】

【発明が解決しようとする課題】ところで、現在の汎用コンピュータで実行されているプログラム、特に大規模なプログラムは、複数のモジュールに分割して作成されている。そして、1のプログラムモジュールが他のプログラムモジュールを呼び出しながら、全体としてのプログラムの実行を進めていくようになっている。こうしてプログラムモジュール別に開発を進めたり、各プログラムモジュールを部品として利用したりすることにより、プログラムの開発期間を短縮することができる。

【0006】しかしながら、上記した従来のF P G Aを用いた演算システムでは、ハードウェアとしてのモジュール分割は考えられていても、ソフトウェアとしてのモジュール分割は考えられていなかった。つまり、ソフトウェアとして1のプログラムモジュールから他のプログラムモジュールを呼び出し、呼び出したプログラムモジュールの実行を終了した後、元のプログラムモジュールに復帰するというように、複数のプログラムモジュールを適時実行していくことにより大規模プログラムの実行を可能とする仕組みは考えられていなかった。

【0007】また、ソフトウェアとしてのモジュール分割を行うとしても、プログラムモジュールを呼び出すごとにF P G Aのコンフィギュレーションを逐次書き換えるようにした場合、書き換えの時間が長いため、オーバーヘッドの多い非効率的な演算システムとなってしまう。

【0008】このため、従来のF P G Aを用いた演算システムで実行可能なプログラムは、実質的に1のみのモジュールで作成されたプログラムでなくてはならないという制約があった。つまり、大規模プログラムの実行が事実上不可能で、その適用範囲は限られるという問題があった。

【0009】本発明は、上記した従来技術の問題点を解消するためになされたものであり、汎用のC P Uを用いることなく、複数のプログラムモジュールからなる大規模プログラムの効率的な実行をハードウェアで直接的に実現した演算システムを提供することを目的とする。

#### 【0010】

【課題を解決するための手段】上記目的を達成するため、本発明の演算システムは、ルックアップテーブルを含んだ複数のプログラムモジュールを各々記憶する複数のプログラム記憶手段と、各前記プログラム記憶手段のいずれかを選択する選択手段と、複数の論理回路を含み、前記選択手段に選択されたプログラム記憶手段に記憶されているプログラムモジュール中の命令に従った信号を前記複数の論理回路の1以上に入力することで、該

プログラムモジュールに応じた演算を実行する論理演算手段と、を備えることを特徴とする。

【0011】上記演算システムでは論理演算手段のコンフィギュレーションの書き換えによるオーバーヘッドがほとんど生じないため、演算が効率的になり、従って高速になる。

【0012】なお、前記選択手段は、たとえば、シフトレジスタより構成されればよい。また、各前記プログラム記憶手段が、プログラムモジュールを各々書き換え可能に記憶するものである場合、前記選択手段は、所定のプログラム記憶手段が記憶するプログラムモジュールを選択するものであって、各前記プログラムモジュールを、各前記プログラム記憶手段相互間でサイクリックに転送させるものであってもよい。

【0013】各前記プログラム記憶手段は、プログラムモジュールを各々書き換え可能に記憶するものであってもよい。この場合、前記演算システムは、前記プログラム記憶手段にプログラムモジュールをロードするロード手段を更に備えるものとすれば、プログラムモジュールのデータ量の総量が、プログラム記憶手段の記憶容量より大きくても、このプログラムモジュールをすべて利用することが可能になる。

【0014】前記演算システムは、前記論理演算手段の内部状態を退避する退避手段と、所定の条件が成立した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、他のプログラムモジュールを前記ロード手段にロードさせ、該他のプログラムモジュールを記憶したプログラム記憶手段を前記選択手段に選択させ、該他のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、元のプログラムモジュールに応じた演算の実行に復帰させる制御手段と、を更に備えていてもよい。このような構成を有する演算システムは、メモリにロードするプログラムモジュールを切り替えるときに、論理演算手段の内部状態を退避し、また、復元する仕組みを備えている。このため、複数のプログラムモジュールからなる大規模なプログラムも、メモリにロードするプログラムを切り替え、論理演算手段中の論理回路に入力する信号を変えていくことで、ハードウェア的に高速に演算を実行していくことができる。

【0015】上記演算システムにおいて、前記複数のプログラムモジュールのうちの少なくとも一部のプログラムモジュールは、他のプログラムモジュールを呼び出す機能を含むものであってもよい。この場合において、上記演算システムは、前記論理演算手段で演算を実行しているプログラムモジュール中の命令における他のプログラムモジュールの呼び出しを検出する呼び出し検出手段をさらに備えるものと/orすることができ、前記制御手段は、前記呼び出し検出手段が他のプログラムモジュールの呼び出しを検出した場合に、前記論理演算手段の内部

状態を前記退避手段に退避すると共に、呼び出し先のプログラムモジュールが前記プログラム記憶手段に記憶されていないときは該呼び出し先のプログラムモジュールを前記ロード手段にロードさせ、該呼び出し先のプログラムモジュールを記憶しているプログラム記憶手段を前記選択手段に選択させ、該呼び出し先のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、呼び出し元のプログラムモジュールに応じた演算の実行に復帰させるものとすることができる。

【0016】ここで、前記制御手段によって実行が切り替えられるプログラムモジュール間において引数を受け渡すための引数受け渡し手段をさらに備えていてよい。

【0017】これらの仕組みをさらに備えることによって、モジュールの呼び出しを含む大規模なプログラムをハードウェア的に高速に実行することが可能となる。

【0018】上記演算システムにおいて、前記退避手段は、先入れ後出し方式のスタックによって構成されたものとすることができる。

【0019】このようなスタックで構成される退避手段により、例えば、別のプログラムモジュールから呼び出されたプログラムモジュールが、さらに他のプログラムモジュールを呼び出すといった処理も可能となる。また、あるプログラムモジュールが、そのプログラムモジュール自身を呼び出す再帰型のプログラムを実行することも可能となる。

【0020】上記演算システムにおいて、前記プログラム記憶手段に記憶された各プログラムモジュールは、具体的には、たとえば、前記論理演算手段を構成する論理回路に入力する信号に応じたコードによって構成されている。

【0021】なお、各プログラムモジュール中の命令を構成するコードは、ハードウェア記述が可能な言語で記述されたソースプログラムをコンパイルすることによって得ることができる。この場合、モジュール別にソースプログラムを開発したり、モジュールの部品としての利用が可能となり、プログラムの開発期間を短縮することが可能となる。

## 【0022】

【発明の実施の形態】以下、添付図面を参照して、本発明の実施の形態について説明する。

【0023】図1は、この実施の形態にかかる演算システムの構成を示すブロック図である。図示するように、この演算システム1は、FPGAデータ記憶部2と、ローダ3と、FPGAデバイス4とから構成されている。FPGAデータ記憶部2には、複数のモジュールに分かれたFPGAデータモジュール2-1～2-nを記憶している。

## 【0024】FPGAデータモジュール2-1～2-n

は、それぞれハードウェア記述が可能なプログラム言語で記述されている複数のモジュールに分かれたソースプログラム5-1～5-nを、FPGAデバイス4の論理記述を行うべくコンパイラ6がコンパイルしたモジュール毎のデータである。FPGAデータモジュール2-1～2-nはそれぞれ、入力データの値と出力データの値との対応関係を示すルックアップテーブルの形式をとるデータを含んでいる。ソースプログラム5-1～5-nのうちの少なくとも1のモジュールは、他のモジュールのソースプログラム5-1～5-nを呼び出す機能を含んでおり、FPGAデータモジュール2-1～2-nには、他のモジュールの呼び出しのためのデータも含まれている。

【0025】ローダ3は、論理回路等より構成されており、FPGAデータ記憶部2に記憶されたFPGAデータモジュール2-1～2-nをモジュール単位でFPGAデバイス4にロードする。ローダ3によるFPGAデータモジュール2-1～2-nのロードの指示は、演算の実行の開始時に外部から与えられる他、FPGAデバイス4による演算の実行によっても与えられる。ローダ3は、ロードの指示1回につき、この指示に従って、後述のFPGAデータメモリ41a～41dのいずれかに、FPGAデータモジュールをロードする。

【0026】FPGAデバイス4は、ローダ3によってロードされたFPGAデータモジュール2-1～2-nに従って論理構成を行い、外部からの入力データに所定の演算を施して出力データとして出力するもので、シフトレジスタ40と、ゲートアレイ43と、呼び出し検出部44と、退避スタック45と、引数受け渡し部46と、制御部47とを備えている。呼び出し検出部44、退避スタック45、引数受け渡し部46及び制御部47は、論理回路等より構成されている。

【0027】シフトレジスタ40は、4個のFPGAデータメモリ41a～41dを含んでいる。FPGAデータメモリ41a～41dはそれぞれ半導体メモリあるいはフリップフロップ回路などより構成されており、ローダ3がロードしたFPGAデータモジュールを記憶するための記憶領域を有している。

【0028】FPGAデータメモリ41a～41dは、たとえば、それぞれデータバスと、アドレスバスと、ライトイネーブル端子と、リードイネーブル端子とを備える。そして、各自のライトイネーブル端子にアクティブルレベルの信号が供給されているとき、データバスに供給されたデータを、アドレスバスに供給されたアドレスが示す記憶領域に格納する。なお、ローダ3は、ロードの指示を供給されるたびに、たとえば、FPGAデータモジュールをロードする記憶領域を示すアドレスをFPGAデータメモリ41a～41dのアドレスバスに供給し、また、当該ロードの指示が示す特定のFPGAデータメモリのライトイネーブル端子にアクティブルレベルの信号

を供給して、この状態で、FPGAデータメモリ41a～41dに、ロードすべきFPGAデータモジュールを供給するものとする。

【0029】また、シフトレジスタ40は、FPGAデータメモリ41a～41dのうちデータを出力する対象であるFPGAデータメモリ1個を、制御部47より与えられる指示に従って、サイクリックに指定する（すなわち、指定する対象を循環シフトさせる）。そして、指定したFPGAデータメモリが格納しているデータを、データバスを介して出力する。FPGAデータメモリ41a～41dが、上述したデータバス、アドレスバス及びリードイネーブル端子を備え、各自のリードイネーブル端子にアクティブレベルの信号が供給されているとき、アドレスバスに供給されたアドレスが示す記憶領域に格納されているデータを読み出してデータバスを介して出力する場合、シフトレジスタ40は、FPGAデータメモリ41a～41dのうちいずれかのリードイネーブル端子にアクティブレベルの信号を供給するものとする。そして、指示を供給されるたびに、例えば、アクティブレベルの信号の供給先であるイネーブル端子を、計4個のリードイネーブル端子のうちからサイクリックに切り替えるものとする。なお、シフトレジスタ40は、複数のFPGAデータメモリのリードイネーブル端子に同時にアクティブレベルの信号を供給して、当該複数のFPGAデータメモリから一括してデータを読み出すことが可能なように構成されていてもよい。

【0030】ゲートアレイ43は、AND、OR、NOTなどの複数のゲート回路43aと、演算の途中結果を内部状態として保持している複数のフリップフロップ43bとを含んでいる。各ゲート回路43aの出力論理は、FPGAデータメモリ41a～41dのうちシフトレジスタ40により指定されたものに記憶されたFPGAデータモジュールに従って変更される。また、各フリップフロップ43bは、所望のデータを外部から書き込むことができるようになっている。

【0031】呼び出し検出部44は、FPGAデータメモリ41a～41dのうちセレクタ42により選択されたものに記憶されたFPGAデータモジュールに含まれる他のモジュールの呼び出しのためのデータを検出する。退避スタック45は、呼び出し検出部44によって他のモジュールの呼び出しのためのデータが検出されたとき、ゲートアレイ43中のフリップフロップ43bに保持されているデータと、呼び出し元のFPGAデータモジュールの識別データとを退避するためのスタックである。退避の手法としては、例えば、先入れ後出し方式を用いる。

【0032】引数受け渡し部46は、モジュールの呼び出し、復帰の際に呼び出し元と呼び出し先のFPGAデータモジュール間における引数の受け渡しを行うものである。より詳細に説明すると、呼び出しの際に

は、呼び出し元のFPGAデータモジュールに従った演算の途中結果としてフリップフロップ43bの所定のものに保持されていたデータを、呼び出し先のFPGAデータモジュールに従った演算の入力（引数）として与える。復帰の際には、呼び出し先のFPGAデータモジュールに従った演算結果（戻り値）の出力データを、ゲートアレイ43中のフリップフロップ43bの所定のものに書き込む。

【0033】制御部47は、呼び出し検出部44が他のモジュールの呼び出しのためのデータを検出した場合、当該呼び出しのためのデータの前までのFPGAデータモジュールに従った演算の途中結果としてフリップフロップ43bのそれぞれに保持されているデータと、呼び出し元のデータモジュールの識別データとを退避スタック45に退避させると共に、呼び出し先のFPGAデータモジュールに従った演算で使用するデータを保持するフリップフロップ43bのデータを、引数受け渡し部46に一時保持させる。その後、FPGAデータメモリ41a～41dのうち呼び出し先のFPGAデータモジュールを記憶するものがあれば、シフトレジスタ40に、該当するFPGAデータメモリが指定されるまで指定する対象を繰り返し循環シフトさせ、なければ、ロード3にロードさせる。そして、引数受け渡し部46に一時保持したデータをゲートアレイ43に入力データとして与える。

【0034】制御部47は、また、呼び出されたFPGAデータモジュールに従った演算が終了したときに、その出力データを引数受け渡し部46に一時保持させる。その後、退避スタック45に退避された呼び出し元のデータモジュールの識別データに従って、シフトレジスタ40に循環シフトを行わせることにより呼び出し元のFPGAデータモジュールを指定させ、退避スタック45に退避されたデータをフリップフロップ43bに復帰させると共に、引数受け渡し部46に一時保持させたデータをフリップフロップ43bの所定のものに書き込ませる。

【0035】なお、FPGAデバイス4に外部から入力される入力データは、キーボードなどの入力装置から入力されるデータの他、磁気ディスク装置などの外部記憶装置から読み出されたデータであってもよい。また、FPGAデバイス4から外部に出力される出力データは、ディスプレイ装置などの出力装置から出力する他、外部記憶装置に書き込むものであってもよく、さらに、周辺機器を制御するための制御データであってもよい。

【0036】以下、この実施の形態にかかる演算システムにおける動作について、具体的な例に基づいて説明する。ここでは、FPGAデータモジュール2-1が最初にロードされるものとし、FPGAデータモジュール2-nは、FPGAデータモジュール2-nを呼び出すものとする。また、退避スタック45は、先入れ後出し方

式でデータの退避を行うものとする。

【0037】FPGAデータモジュール2-1がFPGAデータメモリ41a～41dのいずれかにロードされ（以下では、理解を容易にするため、FPGAデータメモリ41aにロードされたものとする）、セレクタ42がFPGAデータメモリ41aを選択すると、これに従ったレベルの信号がゲート回路43aに入力され、ゲートアレイ43を構成するゲート回路43aが論理構成される。そして、ゲートアレイ43に外部からの入力データが入力されることによって、FPGAデータモジュール2-1に応じた演算がゲートアレイ43において実行される。

【0038】一方、呼び出し検出部44は、FPGAデータメモリ41aにロードされたFPGAデータモジュール2-1にFPGAデータモジュール2-nを呼び出すためのデータが含まれていることを検出し、その旨を制御部47に通知する。制御部47は、その呼び出しにかかる部分の直前までの演算の途中結果としてフリップフロップ43bに保持されているデータ（ゲートアレイ43の内部状態）を、呼び出し元のFPGAデータモジュール2-1を識別するためのデータと共に退避スタック45の一番上に退避させる。また、フリップフロップ43bに保持されているデータのうちで呼び出し先のFPGAデータモジュール2-nに引数として渡すものを、引数受け渡し部46に一時保存させる。

【0039】その後、制御部47は、FPGAデータモジュール2-nがFPGAデータメモリ41b～41dのいずれかにロードされているか否かを判別する。（具体的には、たとえば、FPGAデータモジュール2-nをロードする旨の指示を既に自らローダ3に供給したか否かを判別する。）そして、ロードされていなければ、ローダ3を制御し、呼び出し先であるFPGAデータモジュール2-nをFPGAデータメモリ41b～41dのいずれかにロードさせ（以下では、理解を容易にするため、FPGAデータメモリ41bにロードされたものとする）、シフトレジスタ40にFPGAデータメモリ41bを指定させる。一方、ロードされていれば、シフトレジスタ40に、FPGAデータモジュール2-nがロードされているFPGAデータメモリ（以下では、理解を容易にするため、FPGAデータメモリ41bがこれに該当するものとする）を指定させる。

【0040】FPGAデータモジュール2-nがFPGAデータメモリ41bにロードされ、シフトレジスタ40がFPGAデータメモリ41bを指定すると、これに従ったレベルの信号がゲート回路43aに入力され、ゲートアレイ43を構成するゲート回路43aが論理構成される。また、引数受け渡し部46に引数として一時保存されたデータが、入力データとしてゲートアレイ43に入力され、FPGAデータモジュール2-nに応じた演算がゲートアレイ43において実行される。

【0041】この演算が終了すると、制御部47は、ゲートアレイ43からの出力データを呼び出し元のFPGAデータモジュール2-1に渡す引数として引数受け渡し部46に一時保存させる。制御部47は、さらに退避スタック45の一番上に退避されたデータを参考することでシフトレジスタ40を制御して循環シフトをおこなわせ、呼び出し元のFPGAデータモジュール2-1を記憶するFPGAデータメモリ41aを再び指定させる。

【0042】呼び出し元のFPGAデータモジュール2-1がロードされたFPGAデータメモリ41aが再び指定されると、制御部47は、退避スタック45の一番上に退避されていた内部状態のデータをフリップフロップ43bのそれぞれに書き戻し、ゲートアレイ43の内部状態を復元させる。さらに、引数受け渡し部46に引数として一時保存されていたデータをフリップフロップ43bの所定のものに書き込む。この状態でゲートアレイ43においてFPGAデータモジュール2-1に従った演算が再開され、最終的な演算結果が outputデータとして出力されることとなる。

【0043】なお、FPGAデータモジュール2-1から呼び出されたFPGAデータモジュール2-nが、さらに他のFPGAデータモジュールを呼び出すものであっても演算を実行することができる。FPGAデータモジュール2-nがさらに他のモジュールを呼び出すことを呼び出し検出部44が検出した場合にも、制御部47は、上記と同じような制御を行うものとすればよい。

【0044】以上説明したように、この実施の形態にかかる演算システムでは、ゲートアレイ43の内部状態（フリップフロップ43bが保持するデータ）を退避スタック45に退避した後に、実行中のモジュールとは異なるFPGAデータモジュールがFPGAデータメモリ41a～41dのいずれかにロードされ、ロードされたFPGAデータメモリの記憶内容が outputされるようにしている。また、退避スタック45に退避した状態をゲートアレイ43に復元してから元のモジュールに復帰することができるようになっている。このため、各FPGAデータモジュールをFPGAデータメモリに適宜ロードしていくことによって、複数のモジュールからなる大規模なプログラムを、各モジュールに対応してゲート回路43a間の論理構成を変化させてハードウェア的に実行することができ、従来のCPUを用いた演算システムに比べて高速に演算を実行することができる。

【0045】さらに、この演算システムはFPGAデータメモリを複数備えており、制御に従って適宜選択される。このため、コンフィギュレーションの書き換えによるオーバーヘッドが生じず、FPGAデータメモリが單一でFPGAデータモジュールの呼び出しがある毎に新たなFPGAデータモジュールがロードし直される構成に比べても演算が効率的で高速になる。

【0046】また、FPGAデータモジュール2-1～2-nのうちの少なくとも1のモジュールが他のモジュールを呼び出すためのデータを含んでいるが、このような他のモジュールの呼び出しを含むFPGAデータモジュールがFPGAデータメモリにロードされた場合に、これを呼び出し検出部44が検出している。そして、この検出結果に基づいて、退避スタック45へのゲートアレイ43の内部状態（フリップフロップ43bが保持するデータ）の退避、引数受け渡し部46を介した引数の受け渡しを行っている。また、呼び出し先のモジュールに従った演算が終了したときに、退避スタック45に退避した内部状態の復元、引数受け渡し部46を介した呼び出し元のモジュールへの引数の受け渡しを行っている。このような仕組みを備えることによって、モジュールの呼び出しを含む大規模なプログラムをハードウェア的に実行することが可能となる。

【0047】また、呼び出し検出部44が他のモジュールの呼び出しを検出したときに、ゲートアレイ43の内部状態（フリップフロップ43bが保持するデータ）を退避するのは、先入れ後出し方式の退避スタックである。このため、他のモジュールから呼び出されたモジュールがさらに他のモジュールを呼び出すようなプログラムを実行することもできる。さらに、実行中のモジュールが自身を呼び出す再帰型のプログラムを実行することもできる。

【0048】さらに、FPGAデータモジュール2-1～2-nは、モジュール分割されたソースプログラム5-1～5-nをそれぞれコンパイラ6によってコンパイルしたものである。以上のような特徴を有することによって、この演算システムにおいて実行すべきプログラムは、モジュール別にソースプログラムの開発を進めたり、ソースプログラムの各モジュールを部品として利用したりすることが可能となり、その開発期間を短縮することができる。

【0049】本発明は、上記の実施の形態に限られず、種々の変形、応用が可能である。以下、本発明に適用可能な上記の実施の形態の変形態様について説明する。

【0050】上記の実施の形態では、ローダ3は、FPGAデータ記憶部2に記憶されたいずれかのFPGAデータモジュール2-1～2-nを、そのままFPGAデータメモリ41a～41dにロードするものとしていた。これに対して、FPGAデータモジュール2-1～2-nがマクロを含み、FPGAデータ記憶部2にマクロデータを記憶させておき、ローダ3がFPGAデータメモリ41a～41dにロードする際に、マクロ展開をするものとしてもよい。

【0051】また、FPGAデータメモリ41a～41dはシフトレジスタ40とは別個のものとして、それぞれRAM(Random Access Memory)等のランダムアクセス可能なメモリより構成されていてもよい。そしてこの

場合、例えば図2に示すように、この演算システムは、シフトレジスタ40に代えてセレクタ42を備えるものとしてもよい。

【0052】この場合、FPGAデータメモリ41a～41dは、たとえば、それぞれデータ／アドレスバスと、ライトイネーブル端子と、リードイネーブル端子とを備えるものとする。そして、各自のライトイネーブル端子にアクティブレベルの信号が供給されているとき、データ／アドレスバスにまず供給されたアドレスが示す記憶領域に、このデータ／アドレスバスに次に供給されたデータを格納する。また、各自のリードイネーブル端子にアクティブレベルの信号が供給されているとき、データ／アドレスバスに供給されたアドレスが示す記憶領域に格納されているデータを読み出し、このデータ／アドレスバスを介して出力するものとする。FPGAデータメモリ41a～41dがこのような構成を有している場合、ローダ3は、ロードの指示を供給されるたびに、FPGAデータメモリ41a～41dのうちこの指示が示すもののライトイネーブル端子にアクティブレベルの信号を供給し、この状態で、FPGAデータメモリ41a～41dに、ロードすべきFPGAデータモジュールを供給すればよい。

【0053】セレクタ42は、シフトレジスタ等より構成されており、FPGAデバイス4による演算の実行によって制御部47より与えられる指示に従い、FPGAデータメモリ41a～41dのうちこの指示が示すもの1個を指定する。（すなわち、ゲートアレイ43や呼び出し検出部44がデータを参照することが可能な状態とする。）

FPGAデータメモリ41a～41dが、上述したデータ／アドレスバス及びリードイネーブル端子を備え、各自のリードイネーブル端子にアクティブレベルの信号が供給されているとき、データ／アドレスバスに供給されたアドレスが示す記憶領域に格納されているデータを読み出してこのデータ／アドレスバスを介して出力する場合、セレクタ42は、指示を供給されるたびに、FPGAデータメモリ41a～41dのうちこの指示が示すもののリードイネーブル端子にアクティブレベルの信号を供給すればよい。

【0054】また、FPGAデータメモリ41a～41dがデータを書き換え可能に記憶するメモリより構成されている場合、FPGAデータメモリ41a～41dは、データの入力及び出力に共用されるデータバスに代えて、書き込む対象のデータを入力するための入力ポートと、読み出したデータを出力するための出力ポートとを、互いに別個のものとして備えるようにしてよい。そしてこの場合、FPGAデータメモリ41a～41dは、ひとつのFPGAデータメモリの出力ポートが他のFPGAデータメモリの入力ポートに接続されるようにして、全体として環をなすように接続されていてもよ

い。具体的には、例えば図3に示すように、FPGAデータメモリ41aの出力ポートOUTがFPGAデータメモリ41bの入力ポートINに接続され、FPGAデータメモリ41bの出力ポートOUTがFPGAデータメモリ41cの入力ポートINに接続され、FPGAデータメモリ41cの出力ポートOUTがFPGAデータメモリ41dの入力ポートINに接続され、FPGAデータメモリ41dの出力ポートOUTがFPGAデータメモリ41aの入力ポートINに接続されていてもよい。

【0055】そして、シフトレジスタ40は、データを出力させる対象のFPGAデータメモリをサイクリックに指定する代わりに、FPGAデータメモリ41a～41dに記憶されているデータ自体をFPGAデータメモリ41a～41d相互間でサイクリックに転送させ、ゲートアレイ43には、所定のFPGAデータメモリが記憶しているデータを供給するものとしてもよい。具体的には、例えば図3に模式的に示すように、シフトレジスタ40がデータバスを介して出力するデータは常にFPGAデータメモリ41aが記憶するデータであるものとし、シフトレジスタ40は、制御部47から指示が1回与えられる毎に、FPGAデータメモリ41aが記憶していた内容をFPGAデータメモリ41bに記憶させ、FPGAデータメモリ41bが記憶していた内容をFPGAデータメモリ41cに記憶させ、FPGAデータメモリ41cが記憶していた内容をFPGAデータメモリ41dに記憶させ、FPGAデータメモリ41dが記憶していた内容をFPGAデータメモリ41aに記憶させる、という動作をFPGAデータメモリ41a～41dに行わせるようにしてもよい。

【0056】また、FPGAデータモジュール2-1～2-nは、ローダ3を介さず予めFPGAデータメモリに格納されることとして、FPGAデータメモリが形成するルックアップテーブル自体をシフトジスタが循環シフトさせるようにしてもよい。この場合、図4に示すように、この演算システムはローダ3を省略してもよく、従って制御部47はローダ3を制御する必要はない。一方、シフトレジスタ40は、n個のFPGAデータモジュールを格納するためのn個のFPGAデータメモリ41-1～41-nを備え、FPGAデータメモリ41-1～41-nを指定の対象として循環シフトさせるものとする。FPGAデータメモリ41-1～41-n構成は、上述のFPGAデータメモリ41a～41dと実質的に同一であるものとする。ただし、FPGAデータメモリ41-1～41-nは、必ずしも記憶内容を書き換え可能である必要はない。なお、図4の構成においても、FPGAデータメモリ41-1～41-nはシフトレジスタ40とは別個のものとしてランダムアクセス可能なメモリより構成されていてもよい。この場合、この演算システムは、例えば、シフトレジスタ40に代え

て、図2に示すものと実質的に同一のセレクタ42を備えていればよい。

【0057】また、図1～図4の構成では、ソースプログラム5-1～5-nをそれぞれコンパイルしたFPGAデータモジュール2-1～2-nを、FPGAデバイス4のFPGAデータメモリ41a～41dに適宜ロードしていくものとしていた。これに対して、ソースプログラム5-1～5-nをそのままロードするようにした演算システムを構成することもできる。図5は、このような場合の演算システムの構成を示す。

【0058】この演算システムでは、ローダ3'は、制御部47'からの指示に基づいて、プログラム記憶部5に記憶されたモジュール別のソースプログラム5-1～5-nを適宜メモリ41a'～41d'にロードし、シフトレジスタ40'は、制御部47'からの指示に基づいて、メモリ41a'～41d'のうち、実行すべきソースプログラムを記憶しているものを指定する。インターフィタ48は、シフトレジスタ40'に指定されたメモリにロードされたソースプログラム中の命令を1命令ずつ順次解釈し、その解釈結果に従ってゲートアレイ43'を構成するゲート回路43aに論理構成を行わせるべく所定の信号を出力する。解釈の結果、他のモジュールのソースプログラムを呼び出す命令であった場合には、その旨を制御部47'に通知する。

【0059】制御部47'は、他のモジュールの呼び出しが通知されると、ゲートアレイ43'の内部状態(フリップフロップ43bに保持されているデータ)と、呼び出し元のソースプログラムのモジュールを識別するためのデータと、次に実行すべき命令を示すデータを退避スタック45に退避すると共に、フリップフロップ43bに保持されているデータのうち呼び出し先のモジュールに引数として渡すものを、引数受け渡し部46に一時保存させる。そして、呼び出し先のソースプログラムが既にロードされているか否かを判別し、ロードされていなければ、ローダ3'に呼び出し先のソースプログラムをロードさせ、シフトレジスタ40'に、呼び出し先のソースプログラムがロードされたメモリを指定させて、引数受け渡し部46に一時保存されたデータを入力データとしてゲートアレイ43'に与える。一方、ロードされれば、呼び出し先のソースプログラムがロードされたメモリをシフトレジスタ40'に指定させて、引数受け渡し部46に一時保存されたデータを入力データとしてゲートアレイ43'に与える。

【0060】また、呼び出し先のソースプログラムに従った演算が終了すると、ゲートアレイ43'からの出力データを呼び出し元のモジュールに渡す引数として引数受け渡し部46に一時保存させる。そして、退避スタック45に退避されたデータに従って、呼び出し元のソースプログラムを記憶するメモリを再びシフトレジスタ40'に指定させ、退避スタック45に退避された内部状

態をフリップフロップ43bに戻し、引数受け渡し部46に一時保存された引数をフリップフロップ43bのうちの所定のものに書き込ませる。そして、退避スタック45に退避されたデータに基づいて呼び出し元のモジュールのソースプログラムに従った演算を再開させる。

【0061】なお、インタプリタ48は、複数のゲート回路の組み合わせによるハードウェアで構成することができ、その出力によってゲートアレイ43'に含まれるゲート回路43aの論理構成を、演算の実行速度にほとんど影響を与えることなく高速に行うことができる。また、ここでのゲートアレイ43'は、ソースプログラム中の各命令を終了したときのデータをフリップフロップ43bの所定のものに保持させることで、各命令を順次実行していくことができる。

【0062】以上のようにインタプリタ48を含む構成とすることによって、ソースプログラム5-1～5-nをモジュール別に順次FPGAデバイス4'にロードしていくことが可能となる。このため、FPGAデバイス4'の構成に合わせたコンパイラがなくても、複数のモジュールからなる大規模なプログラムに従った演算を、ハードウェア的に高速に行うことが可能となる。さらに、図5の演算システムはメモリを複数備えており、制御に従って適宜選択される。このため、コンフィギュレーションの書き換えによるオーバーヘッドが生じず、メモリが單一でFPGAデータモジュールの呼び出しがある毎に新たなソースプログラムがロードし直される構成に比べても演算が効率的で高速になる。

【0063】なお、図5の構成においても、メモリ41a'～41d'はシフトレジスタ40'とは別個のものとして、ランダムアクセス可能なメモリより構成されてもよい。この場合、例えば図6に示すように、この演算システムは、シフトレジスタ40'に代えて、図2の構成におけるセレクタ42と実質的に同一の動作を行うセレクタ42'を備えるものとしてもよい。

【0064】また、メモリ41a'～41d'は図3に示すFPGAデータメモリ41a～41dと実質的に同一の構成を有していてもよく、この場合、シフトレジスタ40'は、図3の構成におけるシフトレジスタ40と実質的に同一の動作を行えばよい。

【0065】また、ソースプログラム5-1～5-nは、ローダ3'を介さず予めメモリに格納することとしてもよい。この場合、図7に示すように、この演算システムはローダ3'を省略できる。一方、シフトレジスタ40'は、n個のソースプログラムを格納するためのn個のメモリ41-1'～41-n'を備え、これらn個のメモリを指定の対象として循環シフトさせるものとする。メモリ41-1'～41-n'の構成は、上述のFPGAデータメモリ41-1～41-nと実質的に同一であるものとする。

【0066】なお、図7の構成においても、メモリ41-1'～41-n'はシフトレジスタ40'とは別個のものとしてランダムアクセス可能なメモリより構成されてもよい。この場合、この演算システムは、例えば、シフトレジスタ40'に代えて、図6に示すものと実質的に同一のセレクタ42'を備えていればよい。

#### 【0067】

【発明の効果】以上説明したように本発明によれば、複数のプログラムモジュールからなる大規模なプログラムであっても、各プログラムモジュールを適時メモリにロードしていく仕組みを有するので、該プログラムに応じた演算の実行をハードウェアで実現することが可能となる。また、コンフィギュレーションの書き換えによるオーバーヘッドの影響がなく演算が効率的となる。

#### 【図面の簡単な説明】

【図1】本発明の実施の形態にかかる演算システムの構成を示すブロック図である。

【図2】本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

【図3】本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

【図4】本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

【図5】本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

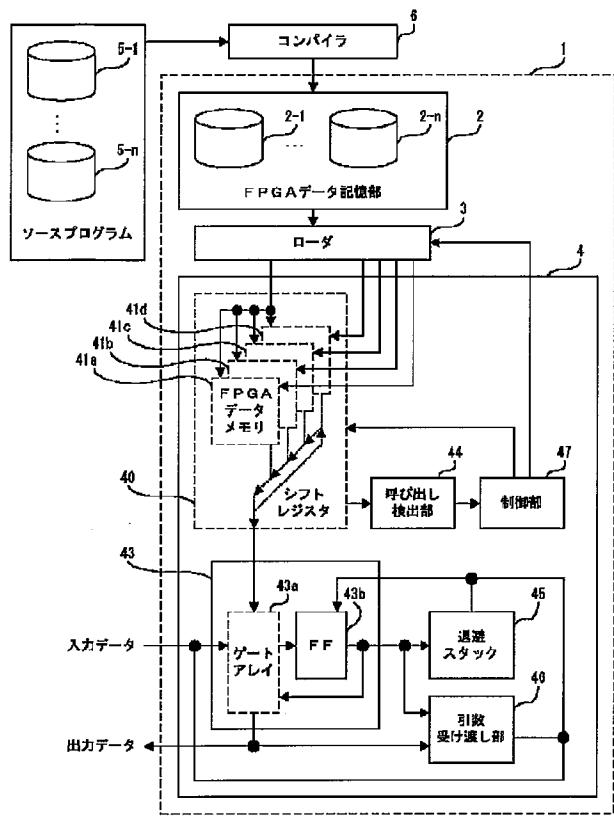
【図6】本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

【図7】本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

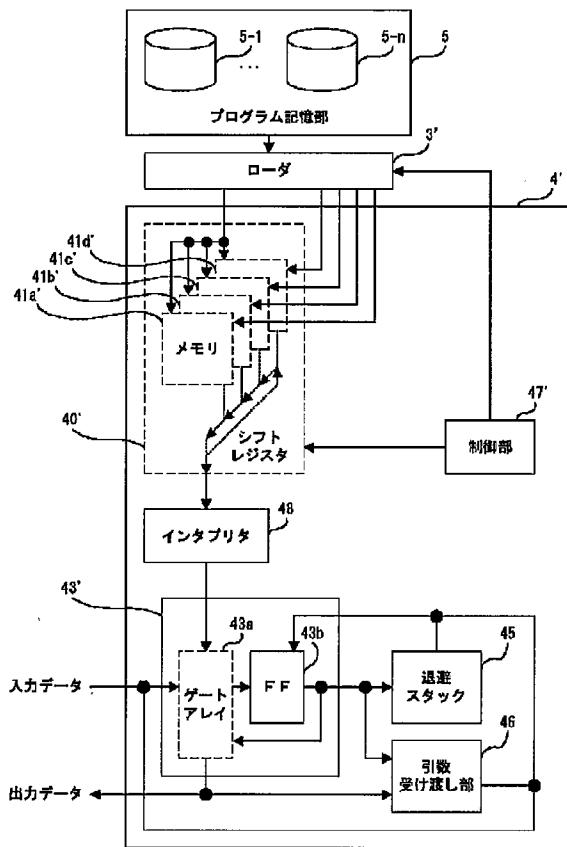
#### 【符号の説明】

- 1 演算システム
- 2 FPGAデータ記憶部
- 3 ローダ
- 4 FPGAデバイス
- 5 コンパイラ
- 6-1～2-n FPGAデータモジュール
- 40 シフトレジスタ
- 41a～41d、41-1～41-n FPGAデータメモリ
- 42 セレクタ
- 43 ゲートアレイ
- 43a ゲート回路
- 43b フリップフロップ
- 44 叫び出し検出部
- 45 退避スタック
- 46 引数受け渡し部
- 47 制御部
- 48 インタプリタ
- 5-1～5-n ソースプログラム

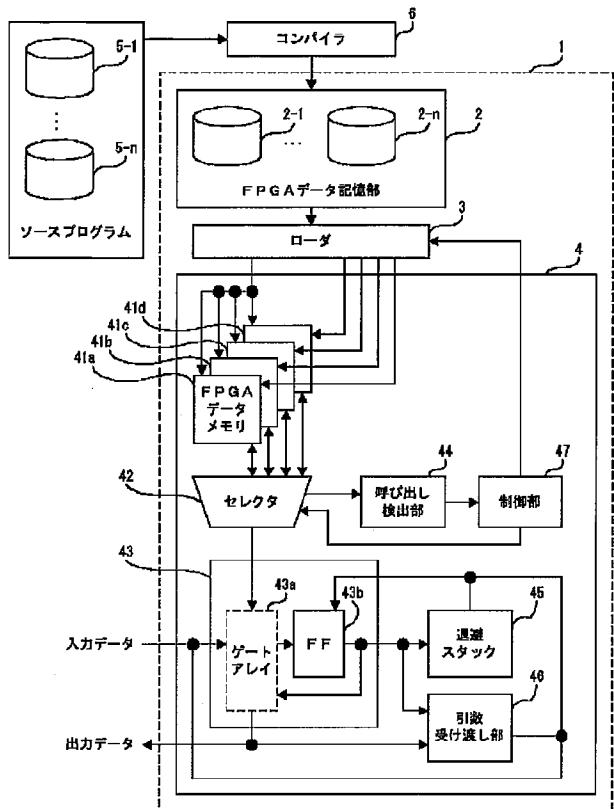
【図1】



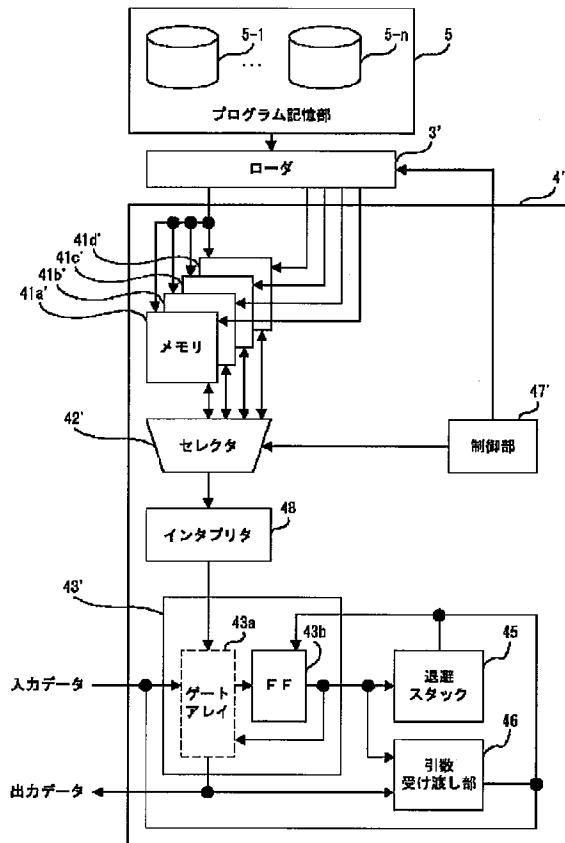
【図5】



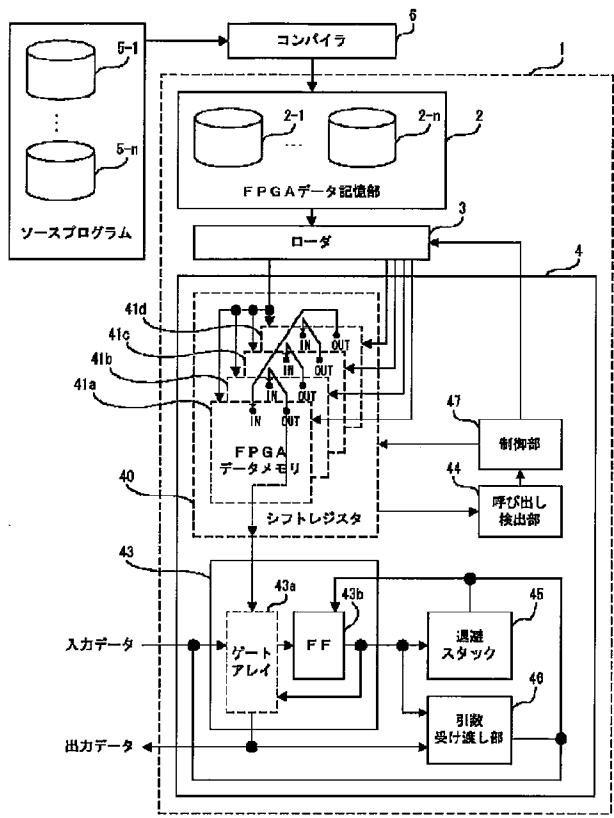
【図2】



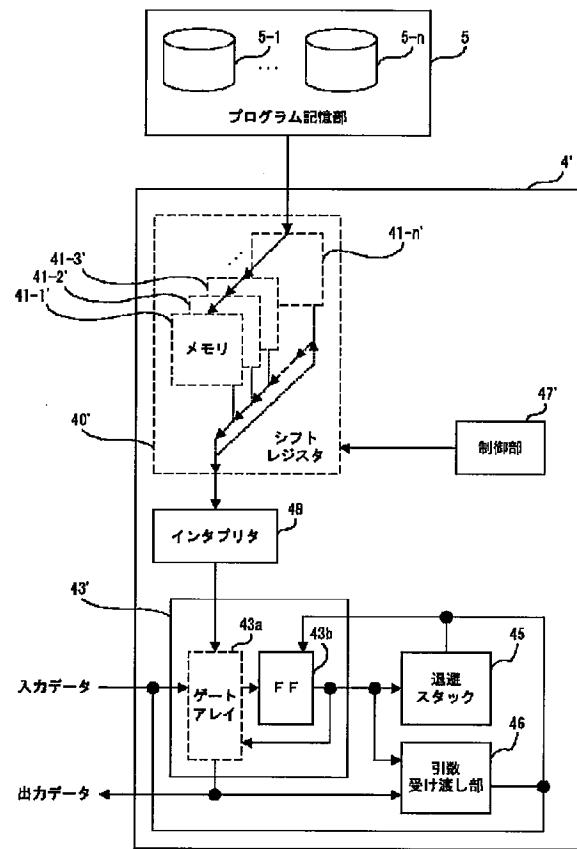
【図6】



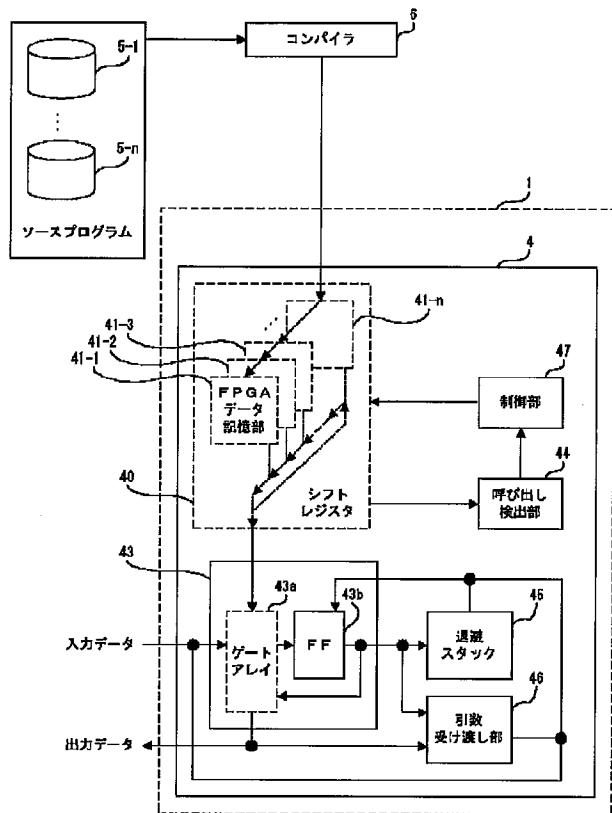
【図3】



【図7】



【図4】



フロントページの続き

(72)発明者 西原 明法

神奈川県川崎市中原区今井仲町374-4-

505

F ターム(参考) 5B033 EA01 EA02

5B076 EB02

5J042 BA03 BA11 CA15 DA01 DA04